

Technical Note

Deep Neural Networks for the Classification of Pure and Impure Strawberry Purees

Zhong Zheng ^{1,2}, Xin Zhang ^{1,3}, Jinxing Yu ^{1,3} , Rui Guo ^{1,3} and Lili Zhangzhong ^{1,3,*}

¹ National Research Center of Intelligent Equipment for Agriculture, Beijing 100097, China; zhongzheng9-c@my.cityu.edu.hk (Z.Z.); zhangx@nercita.org.cn (X.Z.); yujx@nercita.org.cn (J.Y.); guor@nercita.org.cn (R.G.)

² School of Data Science, City University of Hong Kong, Hong Kong, China

³ Key Laboratory for Quality Testing of Hardware and Software Products on Agricultural Information, Ministry of Agriculture, Beijing 100097, China

* Correspondence: lilizhangzhong@163.com

Received: 26 December 2019; Accepted: 21 February 2020; Published: 23 February 2020



Abstract: In this paper, a comparative study of the effectiveness of deep neural networks (DNNs) in the classification of pure and impure purees is conducted. Three different types of deep neural networks (DNNs)—the Gated Recurrent Unit (GRU), the Long Short Term Memory (LSTM), and the temporal convolutional network (TCN)—are employed for the detection of adulteration of strawberry purees. The Strawberry dataset, a time series spectroscopy dataset from the UCR time series classification repository, is utilized to evaluate the performance of different DNNs. Experimental results demonstrate that the TCN is able to obtain a higher classification accuracy than the GRU and LSTM. Moreover, the TCN achieves a new state-of-the-art classification accuracy on the Strawberry dataset. These results indicate the great potential of using the TCN for the detection of adulteration of fruit purees in the future.

Keywords: adulteration detection; deep neural networks; fruit purees; GRU; LSTM; TCN

1. Introduction

The adulteration of fruit purees or juices has long been a serious problem that needs to be carefully considered by manufacturers. This problem arises frequently out of two main reasons. On one hand, the adulteration of fruit purees or juices is profitable since certain fruits command premium prices. For instance, a variety of fruits, such as apple, raspberry, blackcurrant, blackberry, plum, cherry, apricot and grape, are usually used to adulterate strawberry purees. These fruit purees are always cheaper than strawberry purees and could be mixed with strawberry purees to make impure strawberry purees. Impure strawberry purees are then sold as pure strawberry purees for higher profits. On the other hand, the direct detection of the adulteration is quite difficult because of the nuance of flavor and color of pure and impure strawberry purees [1]. To deal with the problem of adulteration detection, a number of quality control methods have been employed, such as high-performance liquid chromatography (HPLC), thin layer chromatography (TLC) enzymatic tests (e.g., sorbitol), and physical tests (e.g., pH) [2]. However, these extensive chemical analyses are always time-consuming and expensive [1], which motivates the adoption of spectroscopic techniques for detecting the adulteration. [3–5] have successfully utilized the Fourier transform infrared (FT-IR) spectroscopy to screen a number of adulterants in a range of food products. Specifically, the mid-infrared spectrum is commonly used for the adulteration detection of fruit purees [6].

The adulteration detection of fruit purees based on spectroscopy is a time series classification (TSC) problem of two classes: the authentic fruit and the adulterated fruit. Based on spectroscopy,

various conventional algorithms have been employed for the adulteration detection of fruit purees. These methods include partial least square regression (PLS) [1], dynamic time warping (DTW) [7], random forest (RF) [8], rotation forest (RotF) [9], etc. Reference [10] compared dozens of conventional algorithms for the adulteration detection of strawberry purees and found that the RotF achieved the best classification accuracy. However, one impediment of conventional methods is they always require manually designed feature extractors, which are usually labor consuming and require specific domain knowledge.

During the last few decades, deep neural networks (DNNs) have recently achieved great success in a number of time series modeling tasks [11–14] and motivate the recent utilization of deep learning models for TSC [15]. Contrary to conventional methods, the biggest advantage of DNNs is the feature extraction could be conducted by the neural network automatically. Two major neural network architectures, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), are commonly adopted for time series analysis. For the RNNs, a number of advanced variants, such as the Long Short Term Memory (LSTM) [16] and the Gated Recurrent Unit (GRU) [17], have been proposed to mitigate the vanishing gradient problem of RNNs due to training on long time series. Among different convolutional architectures, the temporal convolutional network (TCN) has been shown to achieve a comparable performance than RNNs in a number of sequential modeling tasks [18–21]. TCN is a convolutional network specially designed for sequence modeling tasks. The authors of references [19,20] proposed the basic TCN and the advanced TCN, TrellisNet, which achieved state-of-the-art classification accuracy on the Sequential MNIST, Permuted MNIST and Sequential CIFAR-10 datasets. The authors of references [18,21] developed two different variants of the TCN, the stochastic TCN and Wavenet, for the sequence generation.

Motivated by the great success of DNNs, this paper first employs DNNs for the adulteration detection of fruit purees and provides a performance comparison of different DNNs. Three types of DNNs—GRU, LSTM, and TCN—are tested on the Strawberry dataset from the UCR time series classification repository [22]. Experimental results demonstrate that TCN performs best among different DNNs in terms of the classification accuracy on the test dataset. Also, TCN achieves the new state-of-the-art classification accuracy on the Strawberry dataset. This result demonstrates the great potential of using TCN for the adulteration detection of fruit purees.

2. Materials and Methods

In this section, the publicly accessible dataset, Strawberry dataset, from the UCR time series classification repository is first introduced. The training criteria, evaluation metrics, as well as the training descriptions of three deep neural networks for TSC are then described in detail. Next, the network structures of the GRU, the LSTM and the TCN for TSC are introduced in detail, respectively.

2.1. The Strawberry Dataset

The Strawberry dataset contains the mid-infrared spectra of 983 fruit purees, including strawberry purees and non-strawberry purees, such as apple purees and plum purees. This dataset is a time series classification dataset of two classes—strawberry samples and non-strawberry samples. This dataset has been obtained using Fourier transform infrared (FTIR) spectroscopy with attenuated total reflectance (ATR) sampling. Each sample in the Strawberry dataset is a one-dimensional time series of length 235. For all the DNNs, each time series sample of length 235 is fed into the model as a two-dimensional input of size 1×235 . The statistics of the Strawberry dataset are shown in Table 1.

Table 1. Statistics of the Strawberry dataset.

Dataset	Number of Samples
Training dataset	613
Test dataset	370

2.2. Training Criteria and Evaluation Metrics

Given the true label $p \in \{0, 1\}^C$ and classification Bernoulli vector $q \in [0, 1]^C$, the cross entropy loss used to train the DNNs for TSC is shown in Equation (1):

$$L_H = - \sum_{c=1}^C [p_c \log q_c + (1 - p_c) \log(1 - q_c)], \quad (1)$$

where $C = 2$ denotes the total number of classes in the classification problem.

The classification accuracy on the test dataset is used to evaluate the model performance. The formal definition of classification accuracy is shown in Equation (2),

$$\text{classification accuracy} = \frac{N_r}{N} \times 100\%, \quad (2)$$

where N_r and N denote the number of correctly classified samples and the total number of samples in the dataset.

2.3. Training Descriptions

The TCN in the experiment consists of six residual blocks. The dilation factors, d , and the filter size, k , of the dilated causal convolution are $d = 2^i$ for layer i and $k = 6$ for all layers in TCN. The dropout rate used in the TCN is set to 0. Both the GRU and the LSTM are made up of three recurrent layers. To keep approximately the similar size of GRU, LSTM, and TCN, the number of hidden units for the GRU and the LSTM are set to 64 and 56, respectively. In this way, the number of parameters of all DNN models is set to approximately 60,000 to ensure a comparable model complexity. The number of channels of each dilated causal convolution in the TCN is set to 32. All three models are trained by the Adam optimizer with the initial learning rate 0.0001, via minimizing the cross entropy loss between the true label and the predicted classification vector. A grid search is conducted to obtain the best output dropout rate. The total number of training epochs of all models is set to 1000.

In addition, the five-fold cross validation is conducted to find the optimal hyperparameters of all DNNs. Specifically, the output dropout rate varies in the set $\{0.0, 0.1, 0.2\}$. The optimal output dropout rate is selected based on the average classification accuracy on the five validation sub datasets. After the five-fold cross validation, the whole training dataset is used to train the DNNs with the best hyperparameters and the classification accuracies of all models are reported on the test dataset.

2.4. GRU and LSTM for TSC

Both GRU and LSTM are two popular variants of basic RNNs that incorporate gate units into the basic RNN cell to capture the long time dependency in the time series. In the GRU, two gate units, a reset gate r_t and an update gate z_t , are introduced to control the transformation of hidden state h_t at each time t . The control process of hidden states in the GRU is described in Equations (3)–(6):

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (3)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (4)$$

$$\tilde{h}_t = \tanh(W x_t + U(r_t \odot h_{t-1}) + b), \quad (5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (6)$$

As another variant of the RNN, LSTM adopts three gate units, the forget gate f_t , the input gate i_t and the output gate o_t , to control the cell state c_t and hidden state h_t at each time t . The updating of hidden states in LSTM is presented in Equations (7)–(11):

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (7)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (8)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (9)$$

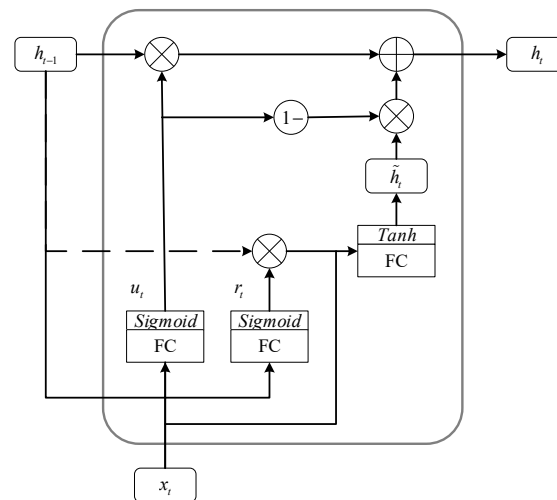
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (10)$$

$$h_t = o_t \odot \tanh(c_t). \quad (11)$$

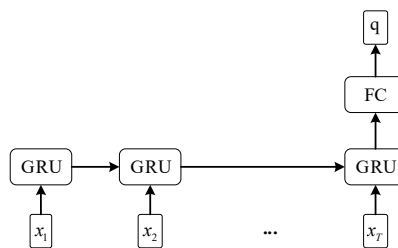
For the TSC problem, the predicted classification vector $q \in \mathbb{R}^C$, where C denotes the total number of classes, is computed by adding a fully connected layer (FC) on top of the hidden state h_T at the last time step T :

$$q = W_q h_T + b_q. \quad (12)$$

In Equations (3)–(12), matrices, W , U , W_r , U_r , W_z , U_z , W_f , U_f , W_i , U_i , W_o , U_o , W_c , U_c , and W_q as well as vectors, b , b_r , b_z , b_f , b_i , b_o , b_c , and b_q , are parameters which need to be optimized. The σ and $\odot$ are utilized to denote and element-wise sigmoid function and the element-wise multiplication, respectively. The $\tanh$ represents the hyperbolic tangent function which serves as the activation function. The schematics of GRU and LSTM for TSC are shown in Figures 1 and 2, respectively.

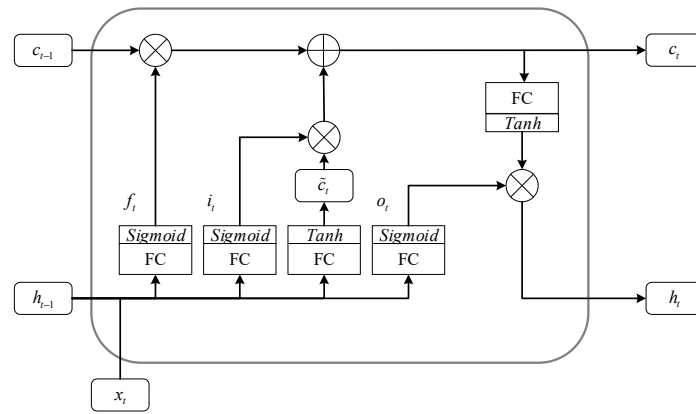


(a) GRU cell

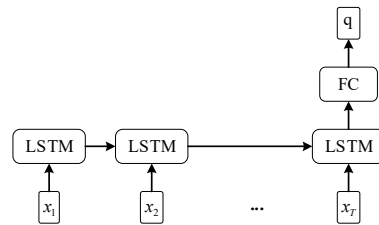


(b) GRU for TSC

Figure 1. The schematic of the Gated Recurrent Unit (GRU) for time series classification (TSC).



(a) LSTM cell



(b) LSTM for TSC

Figure 2. The schematic of the Long Short Term Memory (LSTM) for TSC.

2.5. TCN for TSC

TCN is a specially designed convolutional network for sequence modeling. Formally, given an input sequence $x = (x_1, x_2, \dots, x_T)$, the output sequence $h = (h_1, h_2, \dots, h_T)$ is of the same length as the input sequence. Specifically, the TCN ensures that at each time step t , the output h_t depends only on those inputs from the past: $x_1, x_2, \dots, x_t$. In other words, there is no information leakage from the future at any time. For the purpose of equivalent input and output length, the TCN adopts the 1D fully convolutional network (FCN) architecture [23]. In the FCN, each hidden layer has the same length as the input layer, and the zero padding of length (kernel size -1) is added to keep subsequent layers with the same length as previous ones. In order to prevent information leakage from the future, the TCN employs causal convolutions [19], rather than the normal convolutions. In the causal convolution, an output at time t is convolved only with elements from time t and earlier in the previous layer. In addition, the TCN adopts dilated convolutions [21] to enable an exponentially large receptive field. Different from normal convolutions, dilated convolutions introduce a fixed step between every two adjacent filter taps. Formally, given a 1-D sequence input $x \in \mathbb{R}^n$ and a filter $f: \{0, \dots, k-1\} \rightarrow \mathbb{R}$, the dilated causal convolution operation F on the element s of a sequence x is defined as

$$F(x_s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i}, \quad x_{s \leq 0} = 0 \quad (13)$$

$$h_1, h_2, \dots, h_T = F(x_1), F(x_2), \dots, F(x_T), \quad (14)$$

where d is the dilation factor, k is the filter size, $s - d \cdot i$ counts the number of directions from previous nodes to the current node, and h is the output sequence. Furthermore, TCN also employs the residual block [24], which has repeatedly been utilized to facilitate the construction of very deep networks.

As shown in Figure 3, TCN is constructed by stacking multiple residual blocks. Each residual block consists of two dilated causal convolutions. A fully connected layer is added on top of the last layer at the last time step to output the predicted classification vector.

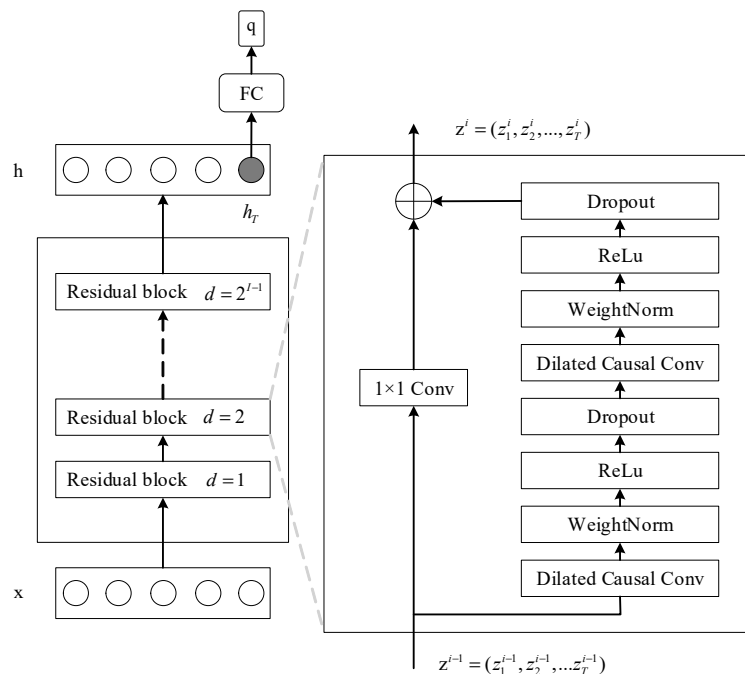


Figure 3. The schematic of the temporal convolutional network (TCN) for TSC.

3. Results and Discussion

The training losses of the GRU, LSTM, TCN, and Multilayer Perceptron (MLP) are illustrated in Figure 4. The comparison of classification accuracy of these three DNNs on the test dataset is then provided in Table 2. Table 3 shows the comparison of the training time of these four DNNs.

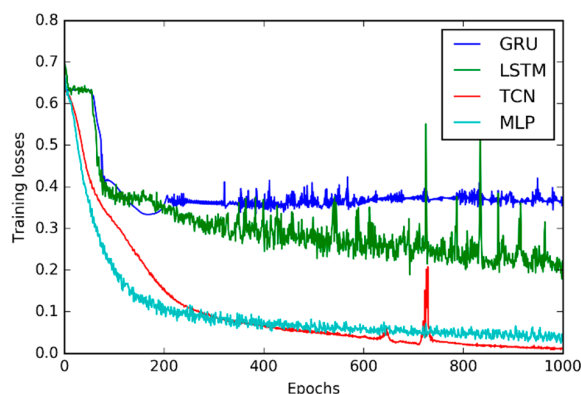


Figure 4. Training losses of the GRU, LSTM, TCN, and MLP.

From Figure 4, it is observable that the TCN converges to lower classification loss than the GRU, LSTM and MLP.

Table 2 shows that the classification accuracy of the GRU and the LSTM is lower than the RotF by [10] as well as the TCN and MLP. This might result from the gradient vanishing problem of the recurrent architectures. On the other hand, the TCN obtains the highest classification accuracy among all the models. The TCN obtains the state-of-the-art classification accuracy of 98.65% on the Strawberry

dataset. This result indicates the great potential of using the TCN for the detection of the adulteration of fruit purees.

Table 2. Classification accuracy (%) of different models on the test dataset.

Model	Classification Accuracy
RotF [2]	97.30
MLP	96.76
GRU	90.54
LSTM	87.84
TCN	98.65

Table 3 shows the comparison of the training time of different DNNs. It is observable that the training of MLP is much faster than the GRU, LSTM and TCN. Furthermore, the training of TCN is much faster than the GRU and LSTM.

Table 3. Training time (in seconds) of different models.

Model	Training Time
MLP	41.44
GRU	253.03
LSTM	250.81
TCN	131.58

4. Conclusions

In this paper, a comparative study of DNNs for the classification of pure and impure strawberry purees was provided. Specifically, three different types of DNNs—GRU, LSTM, and TCN—were implemented for the detection of the adulteration of strawberry purees. These three models were tested on the Strawberry dataset from the UCR time series classification repository. Computational experiments indicated that the TCN obtained a higher classification accuracy than GRU, LSTM, and MLP. Also, in comparison to the best accuracy, 97.30%, obtained by the conventional algorithm RotF [10], the TCN achieved a new state-of-the-art classification accuracy of 98.65% on the Strawberry dataset. These results indicate that it is promising to use the TCN for the detection of the adulteration of fruit purees in the future.

Author Contributions: Conceptualization, Z.Z.; Methodology, Z.Z. and L.Z.; Validation, X.Z. and J.Y.; Writing, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by in part by the National Key R&D Program of China under Grant 2016YFC0403102, in part by the China Agriculture Research System under Grant CARS-23-C06, in part by the Innovation ability construction project of Beijing academy of agriculture and forestry sciences under Grant KJCX20180704, in part by the National Nature Science and Foundation of China under Grant 71801031, in part by the Beijing Natural Science Foundation under Grant 9204028, and in part by the National Key Research and Development Program of China under Grant 2018YFC0810600.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holland, J.; Kemsley, E.; Wilson, R. Use of Fourier Transform Infrared Spectroscopy and Partial Least Squares Regression for the Detection of Adulteration of Strawberry Purees. *J. Sci. Food Agric.* **1998**, *76*, 263–269. [CrossRef]
2. Robards, K.; Antolovich, M. Methods for Assessing the Authenticity of Orange Juice. *A Review. Analyst* **1995**, *120*, 1–28. [CrossRef]
3. Lai, Y.W.; Kemsley, E.K.; Wilson, R.H. Potential of Fourier Transform Infrared Spectroscopy for the Authentication of Vegetable Oils. *J. Agric. Food Chem.* **1994**, *42*, 1154–1159. [CrossRef]

4. Briandet, R.; Kemsley, E.K.; Wilson, R.H. Discrimination of Arabica and Robusta in Instant Coffee by Fourier Transform Infrared Spectroscopy and Chemometrics. *J. Agric. Food Chem.* **1996**, *44*, 170–174. [\[CrossRef\]](#)
5. Petrakis, E.A.; Polissiou, M.G. Assessing Saffron (*Crocus Sativus* L.) Adulteration with Plant-Derived Adulterants by Diffuse Reflectance Infrared Fourier Transform Spectroscopy Coupled with Chemometrics. *Talanta* **2017**, *162*, 558–566. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Defernez, M.; Kemsley, E.K.; Wilson, R.H. Use of Infrared Spectroscopy and Chemometrics for the Authentication of Fruit Purees. *J. Agric. Food Chem.* **1995**, *43*, 109–113. [\[CrossRef\]](#)
7. Rakthanmanon, T.; Campana, B.; Mueen, A.; Batista, G.; Westover, B.; Zhu, Q.; Zakaria, J.; Keogh, E. Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences under Dynamic Time Warping. *ACM Trans. Knowl. Discov. Data TKDD* **2013**, *7*, 10. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
9. Rodriguez, J.J.; Kuncheva, L.I.; Alonso, C.J. Rotation Forest: A New Classifier Ensemble Method. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1619–1630. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E. The Great Time Series Classification Bake off: A Review and Experimental Evaluation of Recent Algorithmic Advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.-A. Deep Learning for Time Series Classification: A Review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [\[CrossRef\]](#)
12. Huang, C.; Wang, L.; Lai, L.L. Data-Driven Short-Term Solar Irradiance Forecasting Based on Information of Neighboring Sites. *IEEE Trans. Ind. Electron.* **2018**, *66*, 9918–9927. [\[CrossRef\]](#)
13. Long, H.; Sang, L.; Wu, Z.; Gu, W. Image-Based Abnormal Data Detection and Cleaning Algorithm via Wind Power Curve. *IEEE Trans. Sustain. Energy* **2019**. [\[CrossRef\]](#)
14. Zhang, S.; Wu, Y.; Che, T.; Lin, Z.; Memisevic, R.; Salakhutdinov, R.R.; Bengio, Y. Architectural Complexity Measures of Recurrent Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 29, NIPS2016, Barcelona, Spain, 5–10 December 2016; pp. 1822–1830.
15. Wang, Z.; Yan, W.; Oates, T. Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1578–1585.
16. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv* **2014**, arXiv:14091259.
18. Aksan, E.; Hilliges, O. STCN: Stochastic Temporal Convolutional Networks. *arXiv* **2019**, arXiv:190206568.
19. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:180301271.
20. Bai, S.; Kolter, J.Z.; Koltun, V. Trellis Networks for Sequence Modeling. *arXiv* **2018**, arXiv:181006682.
21. Van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:160903499.
22. Dau, H.A.; Keogh, E.; Kamgar, K.; Yeh, C.-C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Chen, Y.; Hu, B.; Begum, N.; et al. The UCR Time Series Classification Archive. 2018. Available online: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ (accessed on 22 February 2020).
23. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

